IFW

PTO/SB/30 (11-04)
Approved for use through 07/31/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

PETITION FEE

Under 37 CFR 1.17(f), (g) & (h)

TRANSMITTAL

(Fees are subject to annual revision)

Send completed form to: Commissioner for Patents
P.O. Box 1450, Alexandria, VA 22313-1450

Application Number	**10/849,022**
Filing Date	**May 20, 2004**
First Named Inventor	**H. KIBA, et al**
Art Unit	
Examiner Name	
Attorney Docket Number	**500.43870X00**

Enclosed is a petition filed under 37 CFR §1.102(d) that requires a processing fee (37 CFR 1.17(f), (g), or (h)). Payment of $ 130.00 is enclosed.

This form should be included with the above-mentioned petition and faxed or mailed to the Office using the appropriate Mail Stop (e.g., Mail Stop Petition), if applicable. For transmittal of processing fees under 37 CFR 1.17(i), see form PTO/SB/17i.

Payment of Fees (small entity amounts are NOT available for the petition (fees)

☒ The Commissioner is hereby authorized to charge the following fees to Deposit Account No. 50-1417:

☐ petition fee under 37 CFR 1.17(f), (g) or (h) ☒ any deficiency of fees and credit of any overpayments

Enclose a duplicative copy of this form for fee processing.

☐ Check in the amount of $ _____ is enclosed.

☒ Payment by credit card (From PTO-2038 or equivalent enclosed). Do not provide credit card information on this form.

Petition Fees under 37 CFR 1.17(f): **Fee $400** **Fee Code 1462**

For petitions filed under:
§ 1.53(e) - to accord a filing date.
§ 1.57(a) - to according a filing date.
§ 1.182 – for decision on a question not specifically provided for.
§ 1.183 – to suspend the rules.
§ 1.378(e) for reconsideration of decision on petition refusing to accept delayed payment of maintenance fee in an expired patent.
§ 1.741(b) – to accord a filing date to an application under §1.740 for extension of a patent term.

Petition Fees under 37 CFR 1.17(g): **Fee $200** **Fee code 1463**

For petitions filed under:
§1.12 - for access to an assignment record.
§1.14 - for access to an application.
§1.47 - for filing by other than all the inventors or a person not the inventor.
§1.59 - for expungement of information.
§1.103(a) - to suspend action in an application.
§1.136(b) - for review of a request for extension of time when the provisions of section 1.136(a) are not available.
§1.295 - for review of refusal to publish a statutory invention registration.
§1.296 - to withdraw a request for publication of a statutory invention registration filed on or after the date the notice of intent to publish issued.
§1.377 – for review of decision refusing to accept and record payment of a maintenance fee filed prior to expiration of a patent.
§1.550(c) – for patent owner requests for extension of time in ex parte reexamination proceedings.
§1.956 – for patent owner requests for extension of time in inter partes reexamination proceedings.
§ 5.12 – for expedited handling of a foreign filing license.
§ 5.15 – for changing the scope of a license.
§ 5.25 – for retroactive license.

Petition Fees under 37 CFR 1.17(h): **Fee $130** **Fee Code 1464**

For petitions filed under:
§1.19(g) – to request documents in a form other than that provided in this part.
§1.84 – for accepting color drawings or photographs.
§1.91 – for entry of a model or exhibit.
§1.102(d) – to make an application special.
§1.138(c) – to expressly abandon an application to avoid publication.
§1.313 – to withdraw an application from issue.
§1.314 – to defer issuance of a patent.

Name (Print/Type)	**Colin D. Barnitz**	*Registration No. (Attorney/Agent)*	**35061**
Signature	Colin D Barnitz	*Date*	August 11, 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	**10/849,022**	Confirmation No. 7040
Applicant	:	KIBA, H. et al.	
Filed	:	May 20, 2004	
Titled	:	METHOD AND SYSTEM FOR DATA PROCESSING FOR CONTROLLING A CACHE MEMORY	
TC/AU	:	2186	
Examiner	:	TBD	
Docket No.	:	500.43870X00	
Customer No.:		24956	

PETITION TO MAKE SPECIAL
(ACCELERATED EXAMINATION UNDER MPEP § 708.02(VIII))

MAIL STOP PETITIONS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The Applicants petition the Commissioner to make the above-identified application special in accordance with 37 CFR §1.102(d). In support of this Petition, pursuant to MPEP § 708.02(VIII), Applicants state the following.

(A) REQUIRED FEE

This Petition is accompanied by the fee set forth in 37 CFR § 1.117(h). Payment of the fee has been made in the manner set forth below in Section (G).

08/15/2005 HALI11 00000023 10849022
01 FC:1464 130.00 OP

(B) ALL CLAIMS ARE DIRECTED TO A SINGLE INVENTION

Following the Preliminary Amendment filed on the same date as this paper, claims 1, 3-6 and 8-11 are pending in the application. All the claims of the application are directed to a single invention. If the Office determines that all claims in the application are not directed to a single invention, Applicant will make election without traverse as a prerequisite to the grant of special status in conformity with established telephone restriction practice.

As set forth in independent claims 1, 6 and 11, the invention is generally directed to a method of controlling a cache memory in a storage unit. Under independent claim 1, the invention is a cache control method in a data processing system having a computer for executing a program, and a storage unit having a cache memory for storing data transmitted as a result of execution of said program and a disk device for storing data stored in said cache memory, wherein said storage unit responds to an input of a request for storing data transmitted from said program to store the transmitted data in said cache memory, and responds to an input of a request for flushing transmitted from said program to store, in said disk device, the data stored in said cache memory, wherein said flush request is transmitted from said program to said storage unit at the timing of a check point in a transaction process operated by said program.

Furthermore, under independent claim 6, the invention is a data processing system comprising a computer for executing a program, and a storage unit having a cache memory for storing data transmitted as a result of execution of said program

and a disk device for storing data stored in said cache memory, wherein said storage unit includes: means responsive to an input of a request for storing data transmitted from said program to store the transmitted data in said cache memory; and means responsive to an input of a request for flushing transmitted from said program to store, in said disk device, the data stored in said cache memory, wherein said flush request is transmitted from said program to said storage unit at the timing of a check point in a transaction process operated by said program.

Additionally, under independent claim 11, the invention is data processing program for functioning a data processing system having a computer for executing a program, and a storage unit having a cache memory for storing data transmitted as a result of execution of said program and a disk device for storing data stored in said cache memory, said program causing said storage unit to execute a step of responding to an input of a request for storing data transmitted from said program to store the data transmitted from said program in said cache memory, and a step of responding to an input of a request for flushing transmitted from said program to store, in said disk device, the data stored in said cache memory, wherein said flush request is transmitted from said program to said storage unit at the timing of a check point in a transaction process operated by said program.

(C) PRE-EXAMINATION SEARCH

A pre-examination search has been conducted, directed to the invention as claimed. The pre-examination search was conducted in the following US Manual of Classification areas:

Class	Subclass
707	2, 9, 10, 100-102, 104.1, 200-206
711	100, 111-114
713	150, 153, 200, 201

Furthermore, a keyword search was conducted on the USPTO's EAST database, including the US patent database, the published patent applications database, and the European and Japanese patent abstract databases. In addition, a search for non-patent literature was conducted on the ACM (Association for Computing Machinery) online databases.

(D) REFERENCES DEEMED MOST-CLOSELY RELATED TO THE SUBJECT MATTER ENCOMPASSED BY THE CLAIMS

Based upon a review of the documents located by the search and the documents already of record in the application, the references deemed to be most-closely related to the subject matter encompassed by the claims are listed below. These documents were made of record in the present application by the Information Disclosure Statement filed on August 2, 2005.

Document No.	Inventor
US 6069635	Suzuoki et al.
US 6243809	Gibbons et al.
US 6412045	Dekoning et al.
US 20020138699	Okamura
JP 408095861	Azezaki
JP 410320300	Sato

Because all of the above-listed references are already of record in the present application, in accordance with MPEP § 708.02(VIII)(D), additional copies of these documents have not been submitted with this Petition.

(E) DETAILED DISCUSSION OF THE REFERENCES

Following a brief discussion of the invention, the references deemed most-closely related are discussed below in Section (E)2, pointing out, with the particularity required by 37 CFR 1.111 (b) and (c), how the claimed subject matter is patentable over the teachings of these documents.

1. Discussion of the Invention

The present invention provides a data processing system that includes a computer for executing a program, and a storage unit having a cache memory for storing data transmitted as result of execution of the program, with a disk device in the storage unit for storing data stored in the cache memory. The storage unit responds to an input of a request for storing data transmitted from the program to store the transmitted data in the cache memory. Next, the storage unit responds to an input of a request for flushing transmitted from the program to store, in the disk device, the data stored in the cache memory. It is submitted that the cited references, whether taken individually, or in combination, fail to teach or suggest the invention as claimed in independent claims 1, 6 and 11.

As set forth in claims 1, 6 and 11, a first feature of the invention includes a computer for executing a program, wherein a storage unit has a cache memory for storing data transmitted from the program, and responsive to an input of a request for flushing transmitted from the program, stores, in a disk device, the data stored in the cache memory, wherein the flush request is transmitted from the program to the storage unit at the timing of a check point in a transaction process operated by the program.

As will be discussed in more detail below, the prior art does not teach or suggest, at a minimum, the above-described feature.

2. Discussion of the References Deemed to be Most-Closely Related

The patent to Suzuoki et al., US 6069635, discloses a computer system having a CPU 51, and peripheral devices for providing a desired color lookup table (CLUT). In addition, a cache flash command of a specific command is used for flashing contents to a texture cache and the CLUT. Moreover, the texture cache is flashed automatically (while CLUT is preserved) when a host-to-local or local-to-host transfer is carried out. (See, e.g., Abstract; column 16, lines 4-12; and Figures 1 and 36.) However, unlike the present invention, Suzuoki et al. do not disclose that a storage unit stores transmitted data in a cache memory, or flushes the cache to disk in response to a request from a program. More particularly, Suzuoki et al. do not teach a computer for executing a program, wherein a storage unit has a cache memory for storing data transmitted from the program, and responsive to an input of

a request for flushing transmitted from the program, stores, in a disk device, the data stored in the cache memory, as recited in claims 1, 6 and 11.

The patent to Gibbons et al., US 6243809, discloses a computer system with a non-volatile memory ROM 30. ROM 30 has a flash driver 72, and a flash BIOS interface 74. The flash driver 72 allocates an image buffer 80 and an image header 82 in a memory RAM 16. A non-volatile memory image is flashed to a non-volatile memory 30 (step 180). (See, e.g., Abstract; column 1, lines 30-57; column 6, lines 12-61; and Figures 1, 3A and 8.) However, unlike the present invention, Gibbons et al. do not disclose a storage unit that stores transmitted data in a cache memory, and stores the data to a disk device in response to a request for flushing. More particularly, Gibbons et al. do not teach a computer for executing a program, wherein a storage unit has a cache memory for storing data transmitted from the program, and responsive to an input of a request for flushing transmitted from the program, stores in a disk device the data stored in the cache memory, as recited in claims 1, 6 and 11.

The patent to Dekoning et al, US 6412045, discloses a storage system having a host computer 10 and host adapters 14-16. Each of the host adapters 14-16 connects to each disk array controller 18 and 20 via host SCSI buses 28 and 30. The controller 18 (or controller 20) includes a data processor, subprocessor 32, and a cache memory 33. The cache memory 33 can be divided into a cache memory

area 34 and an alternative cache memory area 36. The host computer 10 generates a cache flushing parameter within the host computer 10 and then transfers the cache flushing parameter to the controller 18 in order to optimize the performance of the host computer. The cache memory of the controller is flushed in accordance with the cache-flushing parameter by operation of the controller. (See, e.g., Abstract; column 1, line 67, through column 2, line 18; column 8, line 14, through column 10, line 21; and Figure 1.) However, unlike the present invention, Dekoning et al. do not disclose that a program on a computer sends a request for flushing. Rather, Dekoning et al. merely teach changing parameters used by the controllers in determining when to flush the cache. More particularly, Dekoning et al. do not teach a computer for executing a program, wherein a storage unit has a cache memory for storing data transmitted from the program, and responsive to an input of a request for flushing transmitted from the program, stores, in a disk device, the data stored in the cache memory, as recited in claims 1, 6 and 11.

The published patent application to Okamura, US 20020138699, discloses in the Description of Prior Art, a computer system having a CPU 63, a cache memory 67 (including a cache section 72), and a main memory 69. The data in the main memory 69 is updated by executing flashing in response to a command from the CPU 63. The CPU 63 performs overwriting of data in the cache section 72 and adds dirty information to an associated entry in the cache section 72. (See, e.g., Abstract; paragraphs 5-8; and Figures 6A-6B.) However, unlike the present invention,

Okamura does not disclose a storage unit that stores transmitted data in the cache memory, and stores the data in a disk device in response to a request for flushing transmitted from a program on a computer. More particularly, Okamura does not teach a computer for executing a program, wherein a storage unit has a cache memory for storing data transmitted from the program, and responsive to an input of a request for flushing transmitted from the program, stores, in a disk device, the data stored in the cache memory, as recited in claims 1, 6 and 11.

The Japanese patent document to Azezaki, JP 408095861, discloses an electronic computer with a cache memory 18a. The electronic computer is equipped with a flash operation block 58. The flash operation block 58 performs a cache flashing operation for writing an updated data of a cache block back to a memory element 12 (by inspecting a cache status and performing a flash execution report DF.) (See, e.g., Abstract; Constitution; and Figure.) However, unlike the present invention, Azezaki does not disclose a storage unit that flushes data in the cache memory to disk in response to a request for flushing transmitted by a program on a computer. More particularly, Azezaki does not teach a computer for executing a program, wherein a storage unit has a cache memory for storing data transmitted from the program, and responsive to an input of a request for flushing transmitted from the program, stores, in a disk device, the data stored in the cache memory, as recited in claims 1, 6 and 11.

The Japanese patent document to Sato, JP 410320300, discloses a storage system having a kernel, a write cache 14, a storage device control driver, and a storage device 13. In addition, the storage device control driver controls a storage device 13. Upon detecting an existence of a power supply interruption request (step 22), the kernel outputs a write cache flashing processing request to the storage device control driver. The driver then receives the processing request and executes processing for flashing a write cache 14 in the storage device 13 (step 24). After completion, the driver informs the kernel of the end of the processing (end information). The kernel then executes the power supply interruption processing (step 26). (See, e.g., Abstract; Solution; and Figure.) However, unlike the present invention, Sato does not disclose a flush request that is transmitted from the program to the storage unit at the timing of a check point in a transaction process operated by the program. More particularly, Sato does not teach a storage unit that, responsive to an input of a request for flushing transmitted from a program, stores, in a disk device, data stored in a cache memory, wherein the flush request is transmitted from the program to the storage unit at the timing of a check point in a transaction process operated by the program, as recited in claims 1, 6 and 11.

(F) CONCLUSION

As demonstrated by the above discussion, the references fail to teach or suggest, at a minimum, a computer for executing a program, wherein a storage unit has a cache memory for storing data transmitted from the program, and responsive

to an input of a request for flushing transmitted from the program, stores, in a disk device, the data stored in the cache memory, wherein the flush request is transmitted from the program to the storage unit at the timing of a check point in a transaction process operated by the program, as recited in claims 1, 6 and 11.

Thus, it is submitted that all of these claims are patentable over the cited references taken individually, or in combination with each other. The remaining claims are dependent claims, claim additional features of the invention, and are patentable at least because they depend from allowable base claims. Accordingly, the requirements of 37 CFR §1.102(d) having been satisfied, the Applicants request that this Petition to Make Special be granted and that the application be examined according to prescribed procedures set forth in MPEP §708.02 (VIII).

The Applicants prepared this Petition in order to satisfy the requirements of 37 C.F.R. §1.102(d) and MPEP §708.02 (VIII). The pre-examination search required by these sections was "directed to the invention as claimed in the application for which special status is requested." MPEP §708.02 (VIII). The search performed in support of this Petition is believed to be in full compliance with the requirements of MPEP §708.02 (VIII); however, Applicants make no representation that the search covered every conceivable search area containing relevant prior art. It is always possible that prior art of greater relevance to the claims may exist. The Applicants urge the Examiner to conduct his or her own complete search of the prior art, and to thoroughly examine this application in view of the prior art cited above and any other prior art that may be located by the Examiner's independent search.

Further, while the Applicants have identified and discussed certain portions of each cited reference in order to satisfy the requirement for a "detailed discussion of the references, which discussion points out, with the particularly required by 37 C.F.R. §1.111(b) and (c), how the claimed subject matter is patentable over the references" (MPEP §708.02(VIII)), the Examiner should not limit review of these documents to the identified portions, but rather is urged to review and consider the entirety of each reference.

(G) FEE PAYMENT (37 C.F.R. 1.17(h))

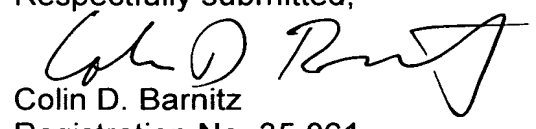
The fee required by 37 C.F.R. § 1.17(h) is to be paid by:

[X] the Credit Card Payment Form (attached) for $130.00.

[ ] charging Account 50-1417 the sum of $130.00.

Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, or credit any overpayment of fees, to the deposit account of MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C., Deposit Account No. 50-1417. A duplicate of this petition is attached.

Respectfully submitted,



Colin D. Barnitz
Registration No. 35,061

MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.
1800 Diagonal Rd., Suite 370
Alexandria, Virginia 22314
(703) 684-1120
Date: August 11, 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	**10/849,022**	Confirmation No. 7040
Applicant	:	KIBA, H. et al.	
Filed	:	May 20, 2004	
Titled	:	METHOD AND SYSTEM FOR DATA PROCESSING FOR CONTROLLING A CACHE MEMORY	
TC/AU	:	2186	
Examiner	:	TBD	
Docket No.	:	500.43870X00	
Customer No.:		24956	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PRELIMINARY AMENDMENT

Sir:

Please amend the application prior to examination, as follows:

Amendments to the Specification begin on page 2 of this paper;

Amendments to the Claims are reflected in the listing of claims which begins on page 4 of this paper;

Amendments to the Drawings begin on page 8 of this paper and include attached replacement sheets; and

Remarks begin on page 9 of this paper.

Appendix is attached following page 9 of this paper including:
Marked Up and Clean Copies of a Substitute Specification; and
Two replacement drawing sheets.

Amendments to the Specification:

Substitute Specification

A Substitute Specification is attached to this Amendment in accordance with 37 CFR § 1.125, with markings showing all changes relative to the original specification. Additionally, a clean version of the Substitute Specification is also attached to this Amendment. Throughout the specification, the terms "flash", "flashed", and "flashing" have been changed to "flush", "flushed", and "flushing", respectively, to correct a typographical error, and the text has been edited to improve readability and correct grammatical errors.

It is asserted that the substitute specification includes no new matter.

Please amend the Abstract as shown on the following page:

ABSTRACT OF THE DISCLOSURE

A data processing system is used which is provided with a computer for executing a program, and a storage unit having a cache memory for storing data transmitted as result of execution of the program and a disk device for storing data stored in the cache memory. The storage unit responds to an input of a request for storing data transmitted from the program to store the transmitted data in the cache memory. Next, the storage unit responds to an input of a request for ~~flashing~~ flushing transmitted from the program to store, in the disk device, the data stored in the cache memory.

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A cache control method in a data processing system having a computer for executing a program, and a storage unit having a cache memory for storing data transmitted as a result of execution of said program and a disk device for storing data stored in said cache memory, wherein

said storage unit responds to an input of a request for storing data transmitted from said program to store the transmitted data in said cache memory, and responds to an input of a request for ~~flash~~flushing transmitted from said program to store, in said disk device, the data stored in said cache memory,

wherein said flush request is transmitted from said program to said storage unit at the timing of a check point in a transaction process operated by said program.

2. (Canceled)

3. (Currently Amended) The ~~A~~ cache control method according to claim 1, wherein each of said data storing request and ~~flash~~flush request includes area identification information for specifying areas in said cache memory, and wherein when said data storing request is inputted, said transmitted data is stored in an area specified by the

area identification information of said data storing request and when said ~~flash~~flush request is inputted, the data stored in the area specified by the area identification information of said data storing request is stored in said disk device.

4. (Currently Amended) The ~~A~~ cache control method according to claim 3, wherein the area of said cache memory is managed as to whether data update occurs in said area or not, and when said ~~flash~~flush request is inputted, data resulting from update of the data stored in the area specified by the area identification information of said data storing request is stored in said disk device.

5. (Currently Amended) The ~~A~~ cache control method according to claim 3, wherein the area identification of said cache memory includes volume identification information and segment identification information.

6. (Currently Amended) A data processing system comprising a computer for executing a program, and a storage unit having a cache memory for storing data transmitted as a result of execution of said program and a disk device for storing data stored in said cache memory, wherein

said storage unit includes:

means responsive to an input of a request for storing data transmitted from said program to store the transmitted data in said cache memory; and

means responsive to an input of a request for ~~flash~~flushing transmitted from said program to store, in said disk device, the data stored in said cache memory,

wherein said flush request is transmitted from said program to said storage unit at the timing of a check point in a transaction process operated by said program.

7. (Canceled)

8. (Currently Amended) The ~~A~~ data processing system according to claim 6, wherein each of said data storing request and ~~flash~~flush request includes area identification information for specifying areas in said cache memory, and wherein when said data storing request is inputted, said transmitted data is stored in an area specified by the area identification information of said data storing request and when said ~~flash~~flush request is inputted, the data stored in the area specified by the area identification information of said data storing request is stored in said disk device.

9. (Currently Amended) The ~~A~~ data processing system according to claim 8, wherein the area of said cache memory is managed as to whether data update occurs in said area or not, and when said ~~flash~~flush request is inputted, data resulting from update of the data stored in the area specified by the area identification information of said data storing request is stored in said disk device.

10. (Currently Amended) The ~~A~~ data processing system according to claim 8, wherein the area identification of said cache memory includes volume identification information and segment identification information.

11. (Currently Amended) A data processing program for functioning a data processing system having a computer for executing a program, and a storage unit having a cache memory for storing data transmitted as a result of execution of said program and a disk device for storing data stored in said cache memory,

said program causing said storage unit to execute a step of responding to an input of a request for storing data transmitted from said program to store the data transmitted from said program in said cache memory, and a step of responding to an input of a request for ~~flash~~flushing transmitted from said program to store, in said disk device, the data stored in said cache memory,

wherein said flush request is transmitted from said program to said storage unit at the timing of a check point in a transaction process operated by said program.

Amendments to the Drawings:

The attached sheets of drawings include changes to FIGS. 1-3. These sheets, which include FIGS. 1-3, replace the original sheets including FIGS. 1-3. In FIGS. 1-3, the terms "flash" and "flashing" have been changed to "flush" and "flushing", respectively, to correct a typographical error.

Attachments: Two Replacement Drawing Sheets

REMARKS

Claims 1, 3-6 and 8-11 have been amended. Claims 2 and 7 have been canceled without prejudice or disclaimer. No claims have been added. Accordingly, claims 1, 3-6 and 8-11 are pending in the application.

Additionally, a Substitute Specification is attached to this Amendment in accordance with 37 CFR § 1.125, with markings showing all changes relative to the original specification. Additionally, a clean version of the Specification is also attached to this Amendment. It is asserted that the substitute specification includes no new matter. Furthermore, the Abstract has been amended to conform to US requirements. Additionally, two replacement drawing sheets are attached.

Entry of this Amendment and examination of the application are requested.

Respectfully submitted,

Colin D. Barnitz
Registration No. 35,061

MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.
1800 Diagonal Rd., Suite 370
Alexandria, Virginia 22314
(703) 684-1120
Date: August 11, 2005

FIG.1

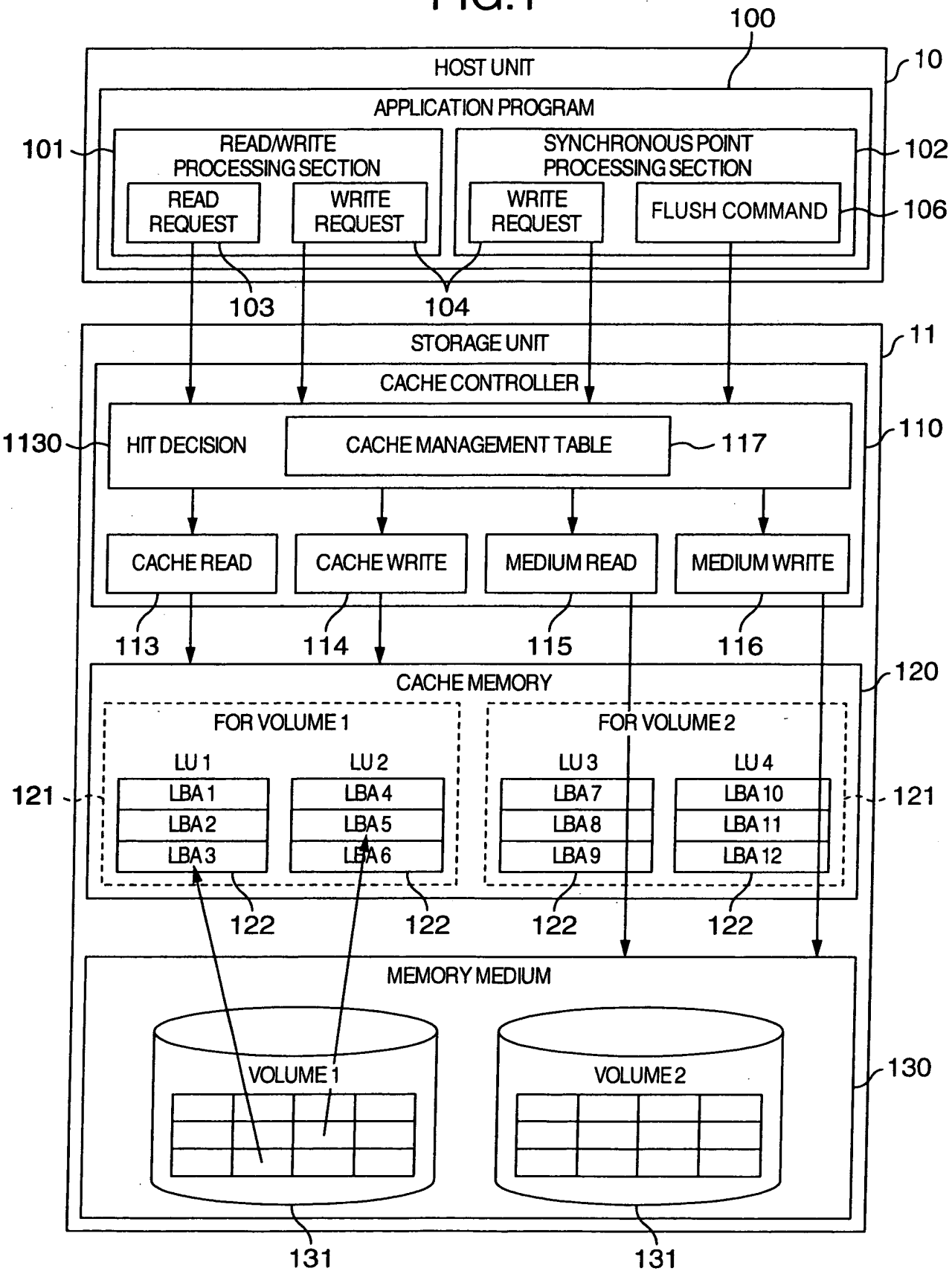

100
HOST UNIT
10
APPLICATION PROGRAM
101
READ/WRITE PROCESSING SECTION
READ REQUEST
WRITE REQUEST
SYNCHRONOUS POINT PROCESSING SECTION
WRITE REQUEST
FLUSH COMMAND
102
106
103
104
STORAGE UNIT
11
CACHE CONTROLLER
1130
HIT DECISION
CACHE MANAGEMENT TABLE
117
110
CACHE READ
CACHE WRITE
MEDIUM READ
MEDIUM WRITE
113
114
115
116
CACHE MEMORY
120
FOR VOLUME 1
FOR VOLUME 2
LU 1
LU 2
LU 3
LU 4
121
LBA 1
LBA 2
LBA 3
LBA 4
LBA 5
LBA 6
LBA 7
LBA 8
LBA 9
LBA 10
LBA 11
LBA 12
121
122
122
122
122
MEMORY MEDIUM
VOLUME 1
VOLUME 2
130
131
131

FIG.2

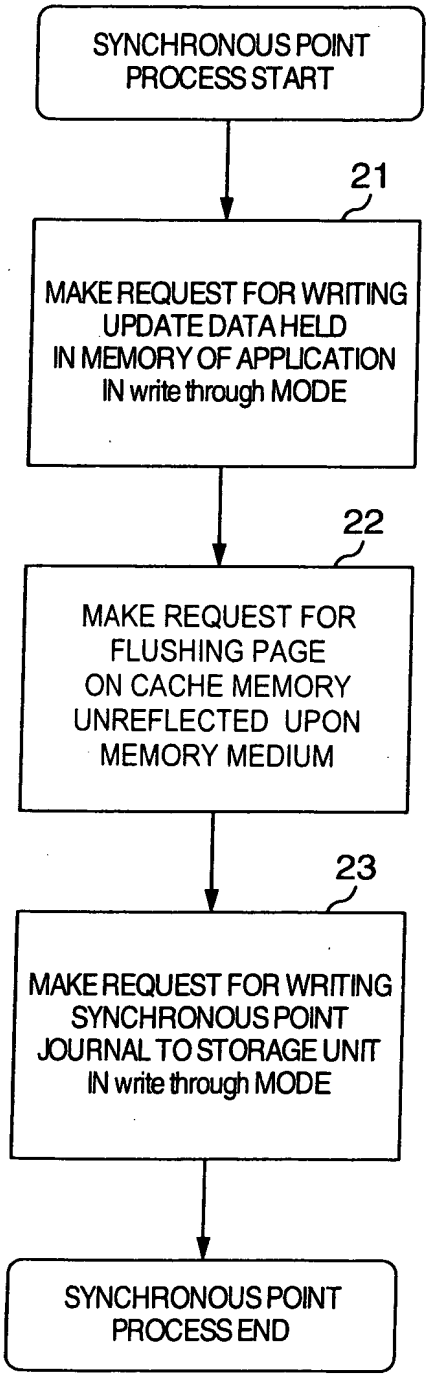
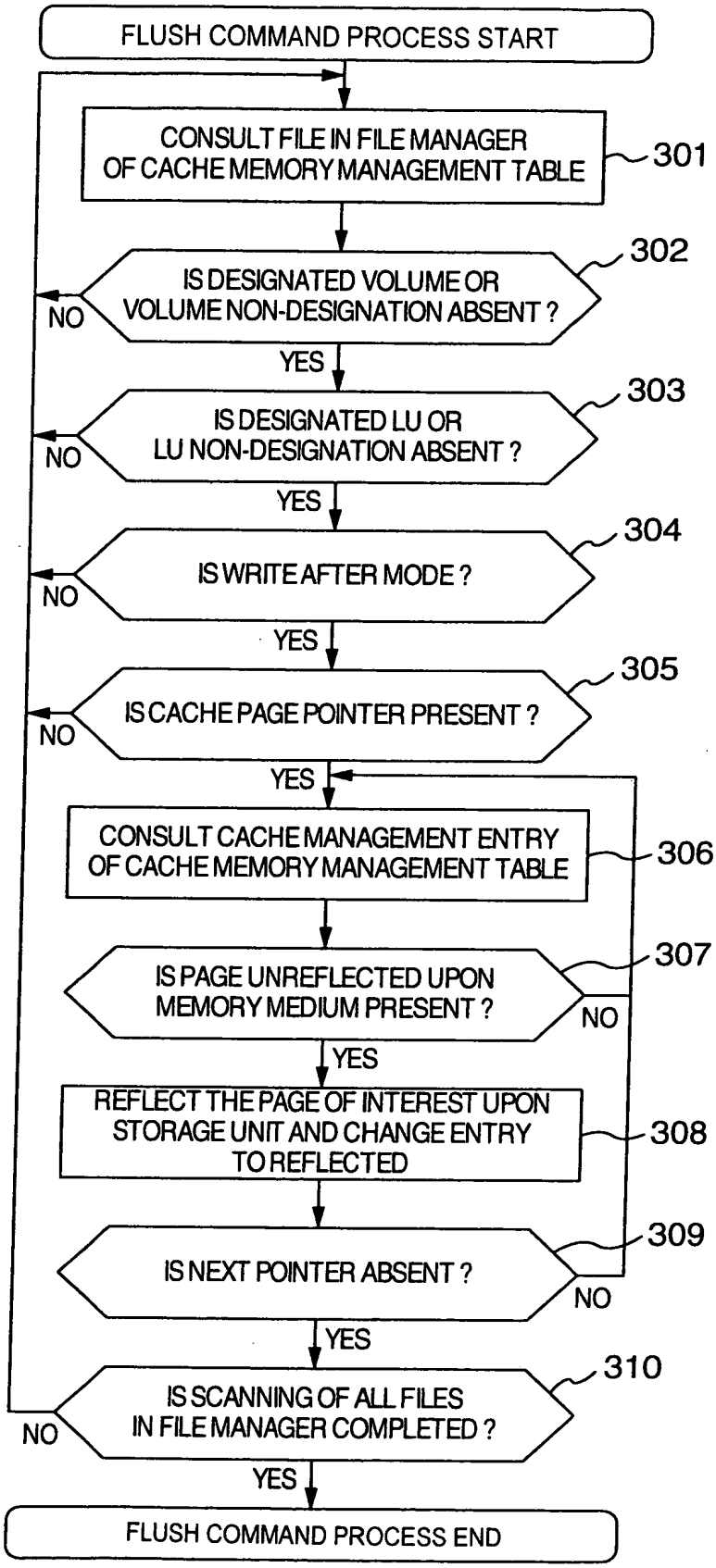

SYNCHRONOUS POINT PROCESS START
21
MAKE REQUEST FOR WRITING UPDATE DATA HELD IN MEMORY OF APPLICATION IN write through MODE
22
MAKE REQUEST FOR FLUSHING PAGE ON CACHE MEMORY UNREFLECTED UPON MEMORY MEDIUM
23
MAKE REQUEST FOR WRITING SYNCHRONOUS POINT JOURNAL TO STORAGE UNIT IN write through MODE
SYNCHRONOUS POINT PROCESS END

FIG.3


FLUSH COMMAND PROCESS START
CONSULT FILE IN FILE MANAGER OF CACHE MEMORY MANAGEMENT TABLE
301
IS DESIGNATED VOLUME OR VOLUME NON-DESIGNATION ABSENT ?
302
NO
YES
IS DESIGNATED LU OR LU NON-DESIGNATION ABSENT ?
303
NO
YES
IS WRITE AFTER MODE ?
304
NO
YES
IS CACHE PAGE POINTER PRESENT ?
305
NO
YES
CONSULT CACHE MANAGEMENT ENTRY OF CACHE MEMORY MANAGEMENT TABLE
306
IS PAGE UNREFLECTED UPON MEMORY MEDIUM PRESENT ?
307
NO
YES
REFLECT THE PAGE OF INTEREST UPON STORAGE UNIT AND CHANGE ENTRY TO REFLECTED
308
IS NEXT POINTER ABSENT ?
309
NO
YES
IS SCANNING OF ALL FILES IN FILE MANAGER COMPLETED ?
310
NO
YES
FLUSH COMMAND PROCESS END

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	**10/849,022**	Confirmation No. 7040
Applicant	:	KIBA, H. et al.	
Filed	:	May 20, 2004	
Titled	:	METHOD AND SYSTEM FOR DATA PROCESSING FOR CONTROLLING A CACHE MEMORY	
TC/AU	:	2186	
Examiner	:	TBD	
Docket No.	:	500.43870X00	
Customer No.:		24956	

SUBMISSION OF SUBSTITUTE SPECIFICATION

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

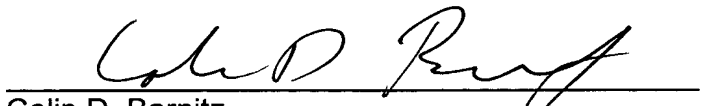
August 11, 2005

Sir:

Attached please find a substitute specification for the above-referenced application. Please note that the attached substitute specification does not contain any new matter. Also attached is a marked-up copy of the specification as originally filed.

It is respectfully requested that any shortage in the fees be charged to the Deposit Account of MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C., Account No. 50-1417 (Case No. 500.43870X00).

Respectfully submitted,



Colin D. Barnitz
Registration Number: 35,061
MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.

CB/jdc
(703) 684-1120

METHOD AND SYSTEM FOR DATA PROCESSING FOR CONTROLLING A CACHE MEMORY

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority from Japanese application JP2004-132479 filed on April 28, 2004, the content of which is hereby incorporated by reference into this application.

BACKGROUND OF THE INVENTION

[0002] The present invention relates to a cache control technique for executing a cache flush process of a storage unit having a cache memory at a timing determined externally of the storage unit.

[0003] A cache control method is available in which, for the purpose of stopping a disk storage unit at the timing that the power supply is interrupted, the contents of a cache memory inside the disk storage unit are forcibly flushed and stored in a magnetic disk or a disk memory device. Such a technique is disclosed in JP-A-10-254780.

SUMMARY OF THE INVENTION

[0004] The conventional technique as above faces a problem that a flush command cannot be applied to the storage unit from a program operating in a data processing unit. In addition, the capacities of cache

memories are increasing, and the conventional technique has a disadvantage that data stored in the cache memory is not stored in the disk memory device until the cache memory is filled up. Consequently, a large amount of the data stored in the cache memory is stored in the disk memory device at the cost of much time when the power supply is interrupted.

[0005] An object of the invention is to achieve sequential write to a memory medium from a cache memory.

[0006] To accomplish the above object, in a synchronous point process of an application program in a host unit, a flush command is applied to a cache memory attached to a storage unit to make the synchronous point process cooperative with a flush process to thereby permit data to be stored sequentially in a memory medium before the cache memory is filled up.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Fig. 1 is a block diagram showing a configuration of a host unit and a storage unit according to an embodiment of the invention.

[0008] Fig. 2 is a flowchart showing a synchronous point process of an application of the host unit in the embodiment of the invention.

[0009] Fig. 3 is a flowchart showing a flush command process by the storage unit in the embodiment of the invention.

[0010] Fig. 4 is a flowchart showing a write request process by the storage unit in the embodiment of the invention.

[0011] Fig. 5 is a diagram showing a data structure of a cache management table in the embodiment of the invention.

DESCRIPTION OF THE EMBODIMENTS

[0012] The present invention will now be described by way of example with reference to the accompanying drawings. Referring first to Fig. 1, a host unit and a storage unit according to an embodiment of the invention are configured as illustrated therein. In the present embodiment, a system comprises a host unit 10 and a storage unit 11. An application program 100 is operating in the host unit 10, and a read/write processing section 101 and a synchronous point processing section 102 are allotted to the application program 100. The read/write processing section 101 issues a read request 103 and a write request 104 to the storage unit 11. The synchronous point processing section 102 issues a write request 104 and a flush command 106 to the storage unit 11. Allotted to the storage unit 11 are a cache controller 110, a cache memory 120 and a memory medium 130. The cache

controller 110 consults a cache management table 117 to carry out a hit decision 1130, a cache read 113 and a cache write 114 as applied to the cache memory 120, and a medium read 115 and a medium write 116 as applied to the memory medium 130. The cache memory 120 is divided into partitions 121 volume by volume, and one partition 121 is fractionated or subdivided into partitions 122, each being in a unit of LU (logical unit). Stored in the LU are LBA's (logical block addresses). The memory medium 130 has a plurality of volumes 131. In the present embodiment, the cache memory 120 is described as being divided into the partitions 121, each being in a unit of volume, and the partitions 122, each being in a unit of LU, but the division into partition 121 in a unit of volume and partition 122 in a unit of LU is not always necessary.

[0013] Referring to Fig. 5, there is illustrated a data structure of the cache management table 117 in the cache controller 110. The cache management table 117 includes a file manager 51 and a cache manager 52. In the file manager 51, file names are held at file 501, volumes storing files are indicated at volume 502, stored LU's are indicated at LU 503 and stored LBA's are indicated at LBA 504. Indicated at mode 505 are available modes of which one is a write through mode in which upon a file write request, write operation is applied to both the memory medium 130 and the cache memory 120, and the other is a write after mode in

which a write to only the cache memory 120 is carried out during a write request and a write operation to the storage medium 130 is carried out during flushing. A head pointer of a list representing a cache management entry 53 used by a file of interest is held at cache pointer 506. The cache manager 52 includes a plurality of cache management entries 53 having information of corresponding cache pages. Next pointer 511 designates the next cache management entry 53 of the file of interest indicated at 501. Page 512 designates a page in the cache memory 120. LBA 513 indicates a store position in the memory medium 130. Held at reflected or unreflected 514 is a status as to whether the cache memory page of interest is reflected upon the memory medium 130 (coincidence of contents) or is not reflected upon the memory medium 130 (non-coincidence of contents).

[0014] Referring to Fig. 2, the flow of a synchronous point process in the synchronous point processing section 102 will be described. A write request 104 is made, in a write through mode, to update data held in a memory of the application program 100 and unreflected upon the storage unit 11 (step 21), a flush command 106 is issued to the storage unit 11 in order to reflect, upon the memory medium 130, a page being on the cache memory 120 and unreflected upon the memory medium 130 (step 22) and a write request 104 is made, in the write through mode, to the storage unit 11

for requesting write of a synchronous point journal which records, in the storage unit 11, completion of the synchronous point process (step 23). In the case of a transaction process, the synchronous point (synchronous timing) corresponds to a checkpoint or the timing for commitment. Accordingly, the application program 100 may be a database management program or a transaction monitor. Alternatively, the application program 100 may be hardware, program or object suitable for notifying the synchronous timing.

[0015] Turning now to Fig. 3, the flow of a flush command process by the cache controller 110 will be described. A file manager 51 (see Fig. 5) of cache management table 117 is scanned and information of a first file at 501 is consulted (step 301). If a condition either for coincidence of a volume commanded by a flush command 106 with a volume at 502 or for non-designation of the volume does not stand, the scanning is returned to the next file at 501 but if the condition stands, the program proceeds to the next step (step 302). Then, if a condition either for coincidence of an LU commanded by the flush command 106 with an LU at 503 or for non-designation of the LU does not stand, the scanning is returned to the next file at 501 but if the condition stands, the program proceeds to the next step (step 303). Subsequently, if a mode at 505 does not coincide with write after, indicating that the file is reflected upon the memory medium 130,

the scanning is returned to the next file at 501, but if the mode at 505 coincides with write after, the program proceeds to the next step (step 304). Then, if a cache pointer at 506 does not point to any cache management entry 53, the scanning is returned to the next file at 501 but if the cache pointer at 506 points to a cache management entry 53, the program proceeds to the next step (step 305). Thereafter, the cache management entry 53 is consulted (step 306). If a page indicated at 512 in the cache management entry 53 has been reflected upon the memory medium 130, the scanning is returned to the next cache management entry 53 but if it is not reflected upon the memory medium 130, the program proceeds to the next step (step 307). Then, the page indicated at 512 in the cache management entry 53 undergoes medium write 116 so as to be written to the memory medium 130 and the cache management entry 53 is changed to reflected (step 308). Subsequently, if the next pointer 511 is present, the scanning is returned to the next cache management entry 53 but if the next pointer 511 is absent, the program proceeds to the next step (step 309). Then, if scanning of all files in the file manager 51 is not completed, the scanning is returned to the next file at 501, but if completed, the flush command process is ended (step 310). The cache controller 110 can be implemented with a program, object or hardware.

[0016] Turning now to Fig. 4, the flow of a write request process by the cache controller 110 will be described. By scanning the file manager 51 of cache management table 117 to retrieve a file at 501 coincident with a file designated by a write request 104, a cache pointer at 506 is acquired (step 401). Then, a cache management entry 53 corresponding to write data is retrieved by collating them with each other (step 402) and if a cache management entry 53 corresponding to the write data is hit, the program proceeds to the next step, but if any cache management entry 53 corresponding to the write data is not hit (step 403), a new cache management entry 53 and a cache memory page are secured and a list of the cache management entry 53 is maintained (step 404). Subsequently, if a mode designated during the write request 104 is write after, the program branches to step 406 (step 405) and the write data undergoes cache write 114 so as to be written to the cache memory 120 (step 406), the cache management entry 53 is changed to unreflected at 514 (step 408) and the program proceeds to step 410. On the other hand, if the mode designated during the write request 104 is write through, the program branches to step 407 (step 405), so that the write data undergoes cache write 114 so as to be written to the cache memory 120, and a medium write 116 is applied to the memory medium 130 (step 407). Then, the cache management entry 53 is changed to reflected

at 514 (step 409) and the program proceeds to step 410. If the collation of all write data with the cache management entry 53 is not completed, the program returns to the collation retrieval, but if completed, the program proceeds to the next step (step 410), and the mode designated during the write request 104 is set at mode 505 (step 411), thus ending the write request process.

[0017] As described above, even in the event that a fault causing the contents of the cache memory of storage unit to be lost takes place, it can be guaranteed that update contents of the application program prevailing up to the synchronous timing (check point or commitment) can be written in the memory medium and, therefore, in the recovery process of the application program after the occurrence of the fault, the recovery start can be determined accurately.

[0018] Further, the sequential cache flush can be carried out by controlling the application program and, therefore, in the event that a power supply failure takes place in the storage unit having a cache memory of a large capacity, much time required for handling the stop process can be avoided, not by the control of the application, but by the flush process of a large amount of update data stored in the cache memory.

[0019] It should be further understood by those skilled in the art that although the foregoing description has been made on embodiments of the

invention, the invention is not limited thereto and various changes and modifications may be made without departing from the spirit of the invention and the scope of the appended claims.

METHOD AND SYSTEM FOR DATA PROCESSING FOR CONTROLLING A CACHE MEMORY

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority from Japanese application JP2004-132479 filed on April 28, 2004, the content of which is hereby incorporated by reference into this application.

BACKGROUND OF THE INVENTION

[0002] The present invention relates to a cache control technique for executing a cache ~~flash~~flush process of a storage unit having a cache memory at ~~the~~ a timing determined externally of the storage unit.

[0003] A cache control method is available in which, for the purpose of stopping a disk storage unit at the timing that the power supply is interrupted, the contents of a cache memory inside the disk storage unit ~~is~~ are forcibly ~~flash~~flushed and stored in a magnetic disk or a disk memory device. Such a technique is disclosed in JP-A-10-254780.

SUMMARY OF THE INVENTION

[0004] The conventional technique as above faces a problem that a ~~flash~~flush command cannot be applied to the storage unit from a program operating in a data processing unit. In addition, the capacities~~y~~ of ~~the~~

cache memories~~y~~ are increasing,~~es~~ and the conventional technique has a disadvantage that data stored in the cache memory is not stored in the disk memory device until the cache memory is filled up. ~~and~~ ~~e~~Consequently, a large amount of the data stored in the cache memory ~~are~~ is stored in the disk memory device at the cost of much time when the power supply is interrupted.

[0005] An object of the invention is to achieve sequential write to a memory medium from a cache memory.

[0006] To accomplish the above object, in a synchronous point process of an application program in a host unit, a ~~flash~~flush command is applied to a cache memory attached to a storage unit to make the synchronous point process cooperative with a ~~flash~~flush process to thereby permit data to be stored sequentially in a memory medium before the cache memory is filled up.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Fig. 1 is a block diagram showing a configuration of a host unit and a storage unit according to an embodiment of the invention.

[0008] Fig. 2 is a flowchart showing a synchronous point process of an application of the host unit in the embodiment of the invention.

[0009] Fig. 3 is a flowchart showing a ~~flash~~flush command process by the storage unit in the embodiment of the invention.

[0010] Fig. 4 is a flowchart showing a write request process by the storage unit in the embodiment of the invention.

[0011] Fig. 5 is a diagram showing a data structure of a cache management table in the embodiment of the invention.

DESCRIPTION OF THE EMBODIMENTS

[0012] The present invention will now be described by way of example with reference to the accompanying drawings. Referring first to Fig. 1, a host unit and a storage unit according to an embodiment of the invention are configured as illustrated therein. In the present embodiment, a system comprises a host unit 10 and a storage unit 11. An application program 100 is operating in the host unit 10, and a read/write processing section 101 and a synchronous point processing section 102 are allotted to the application program 100. The read/write processing section 101 issues a read request 103 and a write request 104 to the storage unit 11. The synchronous point processing section 102 issues a write request 104 and a ~~flash~~flush command 106 to the storage unit 11. Allotted to the storage unit 11 are a cache controller 110, a cache memory 120 and a memory medium 130. The cache

controller 110 consults a cache management table 117 to carry out a hit decision 1130, a cache read 113 and a cache write 114 as applied to the cache memory 120, and a medium read 115 and a medium write 116 as applied to the memory medium 130. The cache memory 120 is divided into partitions 121 volume by volume, and one partition 121 is fractionated or subdivided into partitions 122, each being in a unit of LU (logical unit). Stored in the LU are LBA's (logical block addresses). The memory medium 130 has a plurality of volumes 131. In the present embodiment, the cache memory 120 is described as being divided into the partitions 121, each being in a unit of volume, and the partitions 122, each being in a unit of LU, but the division into partition 121 in a unit of volume and partition 122 in a unit of LU is not always necessary.

[0013] Referring to Fig. 5, there is illustrated a data structure of the cache management table 117 in the cache controller 110. The cache management table 117 includes a file manager 51 and a cache manager 52. In the file manager 51, file names are held at file 501, volumes storing files are indicated at volume 502, stored LU's are indicated at LU 503 and stored LBA's are indicated at LBA 504. Indicated at mode 505 are available modes of which one is a write through mode in which upon a file write request, write operation is applied to both the memory medium 130 and the cache memory 120, and the other is a write after mode in

which a write to only the cache memory 120 is carried out during a write request and a write operation to the storage medium 130 is carried out during ~~flash~~flushing. A head pointer of a list representing a cache management entry 53 used by a file of interest is held at cache pointer 506. The cache manager 52 includes a plurality of cache management entries 53 having information of corresponding cache pages. Next pointer 511 designates the next cache management entry 53 of the file of interest indicated at 501. Page 512 designates a page in the cache memory 120. LBA 513 indicates a store position in the memory medium 130. Held at reflected or unreflected 514 is a status as to whether the cache memory page of interest is reflected upon the memory medium 130 (coincidence of contents) or is not reflected upon the memory medium 130 (non-coincidence of contents).

[0014] Referring to Fig. 2, the flow of a synchronous point process in the synchronous point processing section 102 will be described. A write request 104 is made, in a write through mode, to update data held in a memory of the application program 100 and unreflected upon the storage unit 11 (step 21), a ~~flash~~flush command 106 is issued to the storage unit 11 in order to reflect, upon the memory medium 130, a page being on the cache memory 120 and unreflected upon the memory medium 130 (step 22) and a write request 104 is made, in the write through mode, to the storage unit 11

for requesting write of a synchronous point journal which records, in the storage unit 11, completion of the synchronous point process (step 23). In the case of a transaction process, the synchronous point (synchronous timing) corresponds to a checkpoint or the timing for commitment. Accordingly, the application program 100 may be a database management program or a transaction monitor. Alternatively, the application program 100 may be hardware, program or object suitable for notifying the synchronous timing.

[0015] Turning now to Fig. 3, the flow of a ~~flash~~flush command process by the cache controller 110 will be described. A file manager 51 (see Fig. 5) of cache management table 117 is scanned and information of a first file at 501 is consulted (step 301). If a condition either for coincidence of a volume commanded by a ~~flash~~flush command 106 with a volume at 502 or for non-designation of the volume does not stand, the scanning is returned to the next file at 501 but if the condition stands, the program proceeds to the next step (step 302). Then, if a condition either for coincidence of an LU commanded by the ~~flash~~flush command 106 with an LU at 503 or for non-designation of the LU does not stand, the scanning is returned to the next file at 501 but if the condition stands, the program proceeds to the next step (step 303). Subsequently, if a mode at 505 does not coincide with write after, indicating that the file is reflected upon

the memory medium 130, the scanning is returned to the next file at 501, but if the mode at 505 coincides with write after, the program proceeds to the next step (step 304). Then, if a cache pointer at 506 does not point to any cache management entry 53, the scanning is returned to the next file at 501 but if the cache pointer at 506 points to a cache management entry 53, the program proceeds to the next step (step 305). Thereafter, the cache management entry 53 is consulted (step 306). If a page indicated at 512 in the cache management entry 53 has been reflected upon the memory medium 130, the scanning is returned to the next cache management entry 53 but if it is not reflected upon the memory medium 130, the program proceeds to the next step (step 307). Then, the page indicated at 512 in the cache management entry 53 undergoes medium write 116 so as to be written to the memory medium 130 and the cache management entry 53 is changed to reflected (step 308). Subsequently, if the next pointer 511 is present, the scanning is returned to the next cache management entry 53 but if the next pointer 511 is absent, the program proceeds to the next step (step 309). Then, if scanning of all files in the file manager 51 is not completed, the scanning is returned to the next file at 501, but if completed, the ~~flash~~flush command process is ended (step 310). The cache controller 110 can be implemented with a program, object or hardware.

[0016] Turning now to Fig. 4, the flow of a write request process by the cache controller 110 will be described. By scanning the file manager 51 of cache management table 117 to retrieve a file at 501 coincident with a file designated by a write request 104, a cache pointer at 506 is acquired (step 401). Then, a cache management entry 53 corresponding to write data is retrieved by collating them with each other (step 402) and if a cache management entry 53 corresponding to the write data is hit, the program proceeds to the next step, but if any cache management entry 53 corresponding to the write data is not hit (step 403), a new cache management entry 53 and a cache memory page are secured and a list of the cache management entry 53 is maintained (step 404). Subsequently, if a mode designated during the write request 104 is write after, the program branches to step 406 (step 405) and the write data undergoes cache write 114 so as to be written to the cache memory 120 (step 406), the cache management entry 53 is changed to unreflected at 514 (step 408) and the program proceeds to step 410. On the other hand, if the mode designated during the write request 104 is write through, the program branches to step 407 (step 405), so that the write data undergoes cache write 114 so as to be written to the cache memory 120, and a medium write 116 is applied to the memory medium 130 (step 407). Then, the cache management entry 53 is changed to reflected

at 514 (step 409) and the program proceeds to step 410. If the collation of all write data with the cache management entry 53 is not completed, the program returns to the collation retrieval, but if completed, the program proceeds to the next step (step 410), and the mode designated during the write request 104 is set at mode 505 (step 411), thus ending the write request process.

[0017] As described above, even in the event that a fault causing the contents of the cache memory of storage unit to be lost takes place, it can be guaranteed that update contents of the application program prevailing up to the synchronous timing (check point or commitment) can be written in the memory medium and, therefore, in the recovery process of the application program after the occurrence of the fault, the recovery start can be determined accurately.

[0018] Further, the sequential cache ~~flash~~flush can be carried out by controlling the application program and, therefore, in the event that a power supply failure takes place in the storage unit having a cache memory of a large capacity, much time required for handling the stop process can be avoided, not by the control of the application, but by the ~~flash~~flush process of a large amount of update data stored in the cache memory.

[0019] It should be further understood by those skilled in the art that although the foregoing

description has been made on embodiments of the invention, the invention is not limited thereto and various changes and modifications may be made without departing from the spirit of the invention and the scope of the appended claims.